РУКОВОДСТВО ПРОГРАММИСТА
ДЛЯ НАЧИНАЮЩИХ
ЕЫ2.940.022 РЭ1

СОДЕРЖАНИЕ

1. Компьютер и его подключение 4
2. Клавиатура 5
3. Цифры и буквы. Компьютер в качестве калькулятора . 8
4. Некоторые простые команды 9
5. Элементарное программирование 9
6. Использование магнитофона 11
7. Цвета 13
8. Звук 14
Приложение А.

Краткое описание функций и операторов, реализуемых данной версией языка БЕЙСИК.

1. КОМПЬЮТЕР И ЕГО ПОДКЛЮЧЕНИЕ

Эта брошюра предназначена для тех, кто не знает ничего или почти ничего о компьютерах, и тех, кто знаком с основными компьютерными системами, но хотел бы прочесть инструкцию перед включением устройства в сеть.

Учебником по программированию на языке БЕЙСИК является вторая, более толстая книга, которая называется «Руководство программиста». Эта книга записана на магнитофонную кассету, поставляемую вместе с компьютером «Синтез.2». Человек, незнакомый с компьютерами, должен будет, однако, сначала прочесть и понять эту брошюру.

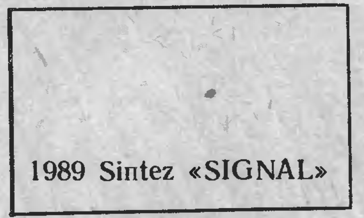
После того, как Вы откроете упаковку с компьютером «Синтез.2», Вы обнаружите в ней:

1. Эту вводную книгу и «Руководство по эксплуатации».
2. Сам компьютер.
3. Блок питания.
4. Шнур для соединения компьютера с телевизором.
5. Шнур для соединения компьютера с магнитофоном.
6. Дополнительные детали.

Компьютер «Синтез.2» дает цветовой сигнал, который обеспечивает цветную картинку (изображение).

Рекомендации по подключению компьютера «Синтез.2» к телевизору, а также порядок подготовки компьютера к работе приведены в «Руководстве по эксплуатации».

После подключения и правильной настройки на экране телевизора появится изображение:



1989 Sintez «SIGNAL»

Эта же картинка появляется и при нажатии на кнопку RESET.

При отсоединении компьютера от сети вся находившаяся в нем информация теряется. Одним из способов сохранения ее является запись на магнитную ленту кассетного магнитофона. Вы можете также покупать записи программ, сделанные другими людьми, и использовать эти программы, загрузив их с помощью магнитофона в компьютер. Подробнее работа с магнитофоном рассматривается в главе 6 данной книги.

Теперь, после того, как компьютер подключен, Вы захотите начать работать с ним. Ниже даются подробные объяснения, как это сделать. Однако Вы, в своем нетерпении, очевидно уже начали нажимать на клавиши и обнаружили, что это изменяет имевшуюся на экране надпись. Не бойтесь: эти Ваши действия не испортят компьютера. Действуйте смело, экспериментируйте! Если Вы ошибетесь, то помните, что можете возвратить компьютер к начальной картинке, нажав на кнопку RESET

2. КЛАВИАТУРА

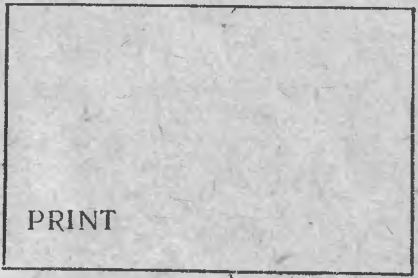
Клавиатура компьютера «Синтез.2» очень похожа на клавиатуру обычной пишущей машинки, однако каждая клавиша может выполнять несколько функций. В обычной пишущей машинке при нормальном режиме работы печатаются маленькие буквы, большие же появляются при нажатии на переключающую клавишу. Клавиатура компьютера «Синтез.2» работает точно так же.

Для того, чтобы помочь Вам определить, в каком режиме находится клавиатура, на экране телевизора высвечивается буква в негативном изображении (белая на черном фоне). Этот знак называется «курсор».

Сразу же после включения компьютера на экране телевизора появляется начальная картинка, описанная в предыдущей главе. Нажатие на любую клавишу приведет к появлению на экране слова, называемого ключевым. Так это делается потому, что компьютер ждет от Вас команды, что ему делать, а все команды начинаются с ключевых слов. Режим ввода ключевого слова индицируется на экране курсором с буквой К.

В отличие от других типов компьютеров, «Синтез.2» позволяет вводить ключевые слова нажатием только одной клавиши.

Например, если сразу же после включения компьютера, нажать на клавишу Р, на экране высветится ключевое слово PRINT на экране появится следующая картинка:



Теперь курсор изменился на букву «L», так как компьютер ожидает буквенного ввода. Напечатайте слово «HELLO ».

Для того, чтобы ввести заглавную букву нужно нажать клавишу CAPS SHIFT и клавишу с нужной буквой. Если на экране высвечивается уже какой-либо другой текст, то нажмите на кнопку RESET и повторите все ранее описанные действия заново.

Команда, начинающаяся со слова PRINT, указывает компьютеру, что нужно напечатать на экране буквы, помещенные между двумя знаками кавычек. Для того, чтобы компьютер приступил к выполнению данной команды, нужно нажать на клавишу ввода ENTER. После того, как это будет сделано, на экране появится слово HELLO и некоторые другие буквы. Мигающий вопросительный знак указывает на то, что где-то в тексте допущена ошибка. Если это так, то упражнение нужно начать сначала. Сообщение «ОК» внизу экрана является подтверждением того, что операция выполнена правильно.

Обратите внимание на то, что буква О и цифра 0 (нуль) обозначаются разными символами. Это важно запомнить. Цифра 0 всегда перечеркивается наклонной линией. Компьютер всегда будет воспринимать литеру О как букву, а не цифру. Аналогично цифра 1 и маленькая буква l (L) не одно и то же, т.е. они не могут взаимозаменяться.

Так как очень важно понять и усвоить режим работы клавиатуры, полезно будет еще раз кратко повторить вышесказанное.

Мигающая буква L называется курсором. Он показывает то место на экране, где компьютер, с вашей помощью поместит следующий написанный знак. Курсор не всегда обозначается как L. Если отключить и вновь включить компьютер (или нажать на кнопку RESET, а потом нажать на клавишу ENTER), то на экране появится курсор K. Буквы, используемые в курсоре, показывают вам, как компьютер будет интерпретировать следующее нажатие на клавишу. В начале строки будет стоять мигающий курсор K, означающий «ключевое слово». Ключевое слово — одно из специальных компьютерных слов, помещающихся в начале команды для того, чтобы дать компьютеру общее представление о том, какого типа команду ему предстоит выполнять. Поскольку компьютер в начале строки ожидает ключевого слова, то, когда Вы нажмете, например, на клавишу P, компьютер принимает решение, интерпретирующее нажатие клавиши не как букву «P», а как ключевое слово PRINT, о чем и уведомляет Вас, мигающий курсор «K». Получив первое ключевое слово, компьютер не ждет второго, поэтому все то, что Вы теперь напечатаете, будет интерпретироваться им как буквы. Чтобы указать на это, компьютер меняет курсор на L. Эти различные состояния работы компьютера называются режимами. В дальнейшем мы будем говорить о работе с ключевыми словами (о работе в режиме K) и о работе с буквами (о работе в режиме L).

Если Вам нужно напечатать много заглавных букв, не держа постоянно нажатой клавишу CAPS SHIFT, Вы можете это сделать, нажав предварительно на клавишу CAPS LOCK. После того, как это будет сделано, курсор L заменится другим мигающим курсором — C (для заглавных букв). Чтобы вернуться обратно к маленьким буквам и курсору L, нужно еще раз нажать клавишу CAPS LOCK.

Если в режиме работы с ключевыми словами Вы нажмете на клавишу CAPS LOCK, то сразу Вы не заметите никакой разницы; она появится после ввода ключевого слова, когда компьютер будет работать не в режиме L, а в режиме C.

Кроме ключевых слов, букв, цифр и различных программных и научных выражений, клавиатура имеет 8 графических знаков. Они совмещены с клавишами цифр от 1 до 8 и могут быть напечатаны на экране так же, как буквы и цифры. Для того, чтобы это сделать, клавиатуру нужно перевести в графический режим. Это осуществляется нажатием на клавишу GRAPH.

Обратите внимание на то, что курсор изменился на G. Вторично нажав на клавишу GRAPH, мы вернемся к работе в режиме L.

Существует еще один режим работы, в который можно перевести клавиатуру, так называемый расширенный. Он обозначается курсором E и переход к нему осуществляется нажатием на клавишу EXTEND. Этот режим позволяет использовать большинство научных и программных функций. Вторичное нажатие на клавишу EXTEND приведет к возвращению к режиму L.

Для исправления ошибочного нажатия на клавишу служит клавиша DELETE. Допустим, что набирая текст ранее описанной команды, вы вместо кавычек напечатали букву P. На экране появится PRINT PHELLO». Компьютер не поймет, что же именно напечатано после ключевого слова PRINT, так как, согласно правилам, он ожидал получить числовую информацию, а вместо нее была введена буква. Свое непонимание компьютер проявит с помощью мигающего вопросительного знака «?».

К счастью Вам не нужно все вводить заново. В нижнем ряду клавиатуры имеются клавиши для перемещения курсора влево, вправо, вниз и вверх и клавиша DELETE для стирания ошибочно напечатанного знака непосредственно перед курсором.

Для того, чтобы исправить допущенную Вами ошибку, нужно нажимать на клавиши ⟨ — или — ⟩ до тех пор, пока курсор не окажется сразу же за буквой P, которую Вы напечатали по ошибке. Если Вы будете держать клавиши нажатыми в течение 1-2 секунд, то они начнут издавать тихий попискивающий звук в процессе работы. Вообще, если Вы задержите на несколько секунд палец на любой клавише, они начнут печатать знаки автоматически. Нажмите на DELETE для стирания неверно напечатанного символа P, а затем напечатайте ″. Тем самым Вы исправили только то, что требовало исправления, не меняя и не набирая заново всю строку. Теперь нужно переместить курсор вправо и вернуться к концу строки. Если Вы снова ошибетесь, то исправьте ошибку этим же способом, но помните, что для этого нужно сначала удалить старый символ, а уже потом печатать новый.

Теперь, если Вы нажмете на клавишу ENTER, компьютер переместит вашу строку в верхнюю часть экрана (или, если там уже имелась какая-либо сохраненная надпись, непосредственно под нее).

Более полное описание клавиатуры и ее функций приведено в «Руководстве программиста» на магнитной ленте.

3. ЦИФРЫ И БУКВЫ. КОМПЬЮТЕР В КАЧЕСТВЕ КАЛЬКУЛЯТОРА

Мы уже знаем, как приказать компьютеру написать буквы и графические знаки с использованием ключевого слова PRINT. Му также знаем, что команда ENTER дается после каждой набранной команды для выполнения ее компьютером.

Цифры обрабатываются на компьютере легче, чем буквы. Если после ввода слова PRINT нажать на цифру 2, на экране появится:

PRINT 2

После нажатия ENTER в верхней части экрана появится просто цифра 2.

Можно также сочетать буквы и цифры:

PRINT 2, «ABC»

Запятая, поставленная между элементами после слова PRINT, разделяет их на 16 колонок при выводе на экран; использование точки с запятой не оставляет пространства.

Инструкция PRINT может использоваться с математическими функциями, в результате чего «Синтез.2» может использоваться и как электронный калькулятор. Например:

PRINT 2 +2

Ответ появится в верхней части экрана. Сравните это с

PRINT «2+2»

Эти две задачи можно объединить в одну для того, чтобы получить что-либо более полезное:

PRINT «2+2=»;2+2

Можно использовать следующие арифметические действия:
«+» — сложение; «—» — вычитание; «*» — умножение;
«/ » — деление.

Можно использовать также отрицательные числа и десятичные дроби (используя точку, а не запятую).

Арифметические действия выполняются в следующей последовательности: сначала умножение и деление, а затем сложение и вычитание. Поэтому выражение 2+3*5 означает, что сначала будет выполнено умножение 3*5=15 и только затем сложение 2+15=17.

Так как операции умножения и деления выполняются в первую очередь, мы говорим, что они имеют более высокий приоритет, чем сложение и вычитание. Между собой операции умножения и деления имеют одинаковый приоритет, что означает, что эти операции выполняются последовательно, одна за другой, при движении вдоль арифметического выражения слева направо.

Такой порядок выполнения арифметических действий неизменен, однако это можно преодолеть, используя скобки; все, что заключено в скобки, подсчитывается в первую очередь, а потом рассматривается как единое число. Например:

```
PRINT 3 * (2+2).
```

дает ответ 3 * 4 = 12.

Более подробно типы данных, виды выражений и математические функции описаны в расширенной документации на магнитной ленте.

4. НЕКОТОРЫЕ ПРОСТЫЕ КОМАНДЫ

Память компьютера может быть использована для хранения разного рода информации. Для этого используется оператор LET (LET — ключевое слово, которому соответствует клавиша L).

```
LET AGE=34
```

При этом определенная часть памяти получает название «AGE» и число 34 заносится туда. Чтобы вывести эту информацию напечатайте

```
PRINT AGE
```

Содержимое ячейки с названием «AGE» легко изменить:

```
LET AGE=56
```

Слово «AGE» является выражением так называемой переменной, так как ее значение можно изменять. Существует также возможность комбинировать отображение на экране текстового сообщения с величиной переменной.

```
PRINT «YOUR AGE IS» ; AGE
```

Но компьютер может больше, чем просто запоминать числа в ячейках с названием. Он может запоминать целые буквенные строки. Для того, чтобы отличать цифровые переменные от буквенных переменных, на конце последних используется знак $.

```
LET A$ = " YOUR AGE IS "
```

Имена буквенных переменных должны иметь только один символ, стоящий рядом с $ (A$). Команда

```
PRINT A$; AGE
```

равносильна предыдущей команде PRINT.

Существуют и другие способы внесения информации в память. Например, команда INPUT в своей простейшей форме сообщает компьютеру, что с помощью клавиатуры должна быть введена определенная информация. Вместо того, чтобы каждый раз набирать слово LET, можно ввести

```
INPUT AGE
```

Как только нажимается клавиша ENTER, на экране появляется мигающий курсор L. Это означает, что компьютер ждет от вас информации. Введите ее и снова нажмите на клавишу ENTER. Переменная AGE приняла заданное вами значение.

5. ЭЛЕМЕНТАРНОЕ ПРОГРАММИРОВАНИЕ

Самое прекрасное в компьютере то, что с его помощью можно программировать. Это означает, что можно ввести в него серию команд, благодаря которым компьютер будет выполнять определенные задания в определенной последовательности.

У каждого компьютера есть свой собственный язык, который позволяет нам общаться с ним. «Синтез.2» использует язык высокого уровня, который называется БЕЙСИК (BASIC). Этот язык был разработан в 1964 году в США. Он очень широко используется в персональных компьютерах, но тем не менее каждый компьютер понимает различные версии БЕЙСИКа, хотя они (эти версии) не сильно отличаются друг от друга. В отличии от других версий БЕЙСИКа, БЕЙСИК компьютера «Синтез.2» не позволяет опускать команду LET при вводе числовых значений переменных.

Существует лимит количества вводимых в компьютер команд. «Синтез.2» сигнализирует об израсходовании этого лимита специальным сигналом типа зуммера.

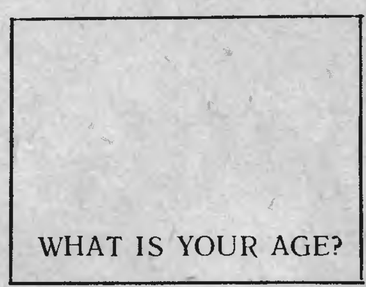
При программировании на БЕЙСИКе необходимо сообщить компьютеру порядок, в котором должны выполняться команды. Поэтому каждая строка инструкции в начале имеет порядковый номер. Обычно нумерация начинается с 10 и увеличивается на 10 с каждой новой строкой. Такой порядок позволяет вводить дополнительные строки, если они были пропущены, или модифицировать программу.

Рассмотрим пример элементарной программы.

Введите следующие инструкции, нажимая на клавишу ENTER в конце каждой строки:

```
10LET B$=«WHAT IS YOUR AGE?»
20 LET A$=«YOUR AGE IS»
30 INPUT (B$); AGE
40 PRINT A$; AGE
```

Ничего не произойдет до тех пор, пока компьютер не получит команду начать работать над программой. Это делается с помощью команды RUN (ключевое слово находится на клавише R). Введите эту команду — и получите:



```
WHAT IS YOUR AGE?
```

Введите Ваш возраст и машина напечатает его на экране.

Если Вы хотите просмотреть программу вторично, то нажмите еще раз клавишу ENTER (или LIST). Мигающая стрелка будет указывать на последнюю введенную строку. Вы можете использовать команду RUN для выполнения программы сколько угодно раз. Когда данная программа больше не нужна, Вы можете ее убрать, использовав команду NEW. Эта команда удаляет программу из памяти компьютера и очищает экран для написания новой программы.

Итак повторим:

— когда Вы набираете команду, которой предшествует номер, то это указание для компьютера, что набранное — не команда, а строка программы. Поэтому она не выполняется, а запоминается для последующего использования;

— «Синтез.2» отображает на экране (делает листинг) все введенные строки программы, отмечая последнюю строку указательной стрелкой . Компьютер не будет выполнять ни одну из этих строк сразу же, а просто поместит их внутри собственной памяти;

— для того, чтобы компьютер начал работать с введенными строками, ему нужно дать команду RUN;

— если вы нажмете на клавишу ENTER, компьютер снова вернется к листингу программы.

6. ИСПОЛЬЗОВАНИЕ МАГНИТОФОНА

Ручной набор программ каждый раз, когда возникает необходимость в их использовании, занятие довольно утомительное. Компьютер «Синтез.2» предусматривает возможность записи программ на магнитную ленту с помощью обычного магнитофона. Если Вы сохраните программу на магнитной ленте, то сможете потом и загрузить ее обратно в компьютер. Для записи программ удобней использовать дешевые переносные кассетные магнитофоны, имеющие счетчик ленты, хотя подходят и другие модели магнитофонов.

Подключите магнитофон к компьютеру при помощи магнитофонного шнура. Этот шнур входит в комплект компьютера и имеет с одного конца одну вилку (которая вставляется в гнездо «М» компьютера), а с другого конца — две вилки (одна вставляется в выходное гнездо магнитофона, другая — во входное гнездо). Подробнее эта операция описана в «Руководстве по эксплуатации».

Для записи программы на ленту используется команда SAVE. Введите в компьютер какую-нибудь программу, а затем напечатайте команду:

```
SAVE «ABCD»
```

«ABCD» — это всего лишь название, которое используется для обозначения программы, находящейся на ленте. Название программы может иметь не более 10 знаков (букв или цифр).

Компьютер должен ответить на вашу команду сообщением:

```
START TAPE THEN PRESS ANY KEY.
```

Пустите ленту и нажмите любую клавишу. На краях телевизионного экрана вы увидете «узор», состоящий из горизонтальных разноцветных полос. В течение 5 секунд красные и бледно-голубые полоски начнут медленно перемещаться вверх. Затем на долю секунды блеснут голубые и желтые полоски. Еще через секунду все станет как обычно. Через 2 секунды — снова «узор» из красных и бледно-голубых полосок. Около одной секунды — опять голубые и желтые.

Пересылаемая информация состоит из двух блоков, и каждый из них имеет свое начало, соответствующее «узору», состоящему из голубых и желтых полос. Первый блок — вступительный, куда входит название программы и другая информация о ней. Второй блок — это сама программа вместе с присутствующими в ней различными переменными. Белый кусочек между этими блоками — это всего лишь промежуток.

Когда компьютер закончит операцию записи (на экране появится сообщение: OK), остановите магнитофон.

Чтобы убедиться в том, что запись произведена успешно, можно сравнить информацию, записанную на пленке, с хранящейся в памяти компьютера при помощи команды VERIFY.

1. Перемотайте кассету на то место, с которого начиналась запись.

2. Напечатайте
VERIFY «ABCD»

3. Включите магнитофон на воспроизведение.

Цвет края экрана будет меняться от красного до голубого до тех пор, пока лента не дойдет до места начала записи. Затем вы увидете тот самый узор, который был во время записывания программы. По окончании «узоров» компьютер остановится. Если на экране появится надпись «OK», то это означает, что программа записана на ленте безошибочно. Если такого не произошло, то это означает, что запись была произведена с ошибкой.

Причиной ошибки может быть либо отсутствие записи на ленте, либо искажение записи. Нужно еще раз проверить правильность подключения магнитофонного шнура, а также уровень записи. Если он очень низкий, то компьютер не сможет верно воспринимать сигнал и на экране не отразится правильный «узор». Если уровень записи очень большой, то сигнал будет искажен. Убедитесь также, что Вы не начали запись на пластиковой части ленты (на ракорде). Если на ленте имеется дефект, то компьютер напечатает сообщение:

R TAPE LOADING ERROR (ошибка в записи на ленте). Это в данном случае означает, что компьютер не может проверить программу. Нужно записывать ее заново.

Теперь допустим, что Вы программу записали и проверили. Нужно ввести ее заново в компьютер. Загрузка программы производится по команде LOAD:
LOAD «ABCD»

Оператор LOAD находится на клавише J.

Если программа проверена безошибочно, то у Вас не должно быть никаких трудностей с загрузкой ее в компьютер.

Операция LOAD убирает старую программу (и переменные) из памяти компьютера перед введением новой программы с ленты.

Когда программа загружена, нужно ввести команду RUN (на клавише R) — и программа будет работать.

Возможно также приобретение программ, ранее записанных на кассетах. Они должны быть написаны специально для компьютера «Синтез.2» или для совместимого с ним компьютера ZX SPECTRUM, так как для различных компьютеров используются разные виды записи программ, и они не взаимозаменяемы.

Если на Вашей кассете записано более одной программы, то загрузить нужную программу вы можете с помощью команды LOAD, указав на ней имя требуемой программы. Например:

LOAD «HELICOPTER»

«HELICOPTER» — имя загружаемой программы.

7. ЦВЕТА

Одной из причин покупки компьютера «Синтез.2» была возможность использовать цвет на телевизионном экране.

Экран разделен на две области. Внешняя часть называется BORDER (бордюр), а средняя — PAPER (бумага). Можно свободно менять цвета обеих этих частей экрана, как с помощью клавиатуры, так и с помощью программы.

Компьютер «Синтез.2» располагает восемью цветами, обозначенными цифрами от 0 до 7. Хотя на первый взгляд цвета подобраны в случайном порядке, однако он фактически представляет собой уменьшающуюся яркость экрана.

Ниже приводится перечень цветов, который дается вместе с соответствующими им цифрами на клавишах:

0	черный
1	голубой
2	красный
3	ярко-красный (или пурпурный)
4	зеленый
5	бледно-голубой или синий
6	желтый
7	белый

При включении компьютера он дает черно-белую картинку. Таким образом, обычная величина цифры для BORDER и PAPER — 7, то есть белый цвет. Цвет любого знака, появляющегося на экране, определяется с помощью команды INK (чернила). Обычно он обозначается цифрой 0, что соответствует черному цвету. Как правило, эти три команды (BORDER, PAPER и INK), (управляющие цветами на экране, изначально устанавливаются самим компьютером. Однако вы можете поменять их значения.

Например, введите команду:

BORDER 2

Если Вы не забыли нажать на ENTER, то края экрана сменят цвет с белого на красный. Это относится также и к нижней части экрана, на которой пишутся команды и инструкции. Затем попробуйте изменить цвет середины экрана, для чего нужно написать:

PAPER 5

Команда PAPER относится к упоминавшейся выше группе команд расширенного режима: она набирается нажатием на клавишу EXTEND, а затем на клавиши SIMBOL SHIFT и C.

После двухкратного нажатия на ENTER центр экрана должен изменить цвет на бледно-голубой. Первое нажатие на клавишу ENTER аннулирует команду PAPER, введенную в компьютер ранее, и только при вторичном нажатии на ENTER следует смена цвета «бумаги».

Команда INK, так же, как и команда PAPER, управляет цветом знаков, высвечивающихся на центральной части экрана (на бумаге).

Естественно, что если INK и PAPER будут обозначены одним и тем же цветом, то на экране ничего не будет видно. Ниже приведен пример программы, демонстрирующей возможное сочетание цветов:

```
10  FOR X=0 TO 7
20  BORDER X
30  PAPER 7 — X: CLS
40  PAUSE 50
50  NEXT X
```

Команда CLS, данная сразу же после PAPER, ведет к тому, что компьютер отражает на экране картинку с использованием нового цвета бумаги. Команда PAUSE задерживает выполнение программы на 1 секунду, благодаря чему мы можем увидеть, что именно происходит.

Существует еще и другие команды, связанные с использованием цветовых возможностей компьютера «Синтез.2». Они подробно рассматриваются в документации на магнитной ленте.

8 ЗВУК

Компьютер «Синтез.2» может генерировать бесконечно разнообразные звуки. Вы сами контролируете их частоту и продолжительность. Команда BEEP используется для указания компьютеру издать звук. Команда BEEP относится к группе команд при работе в расширенном режиме (курсор «E») и набирается нажатием на клавишу «Z».

Нулевая частота для команды BEEP соответствует ноте «ДО» первой октавы. Частота может изменяться в рамках команды BEEP, и может быть получена любая нота, даже если она выражается в полутонах или в долях полутонов ниже или выше нулевой частоты. Если ввести команду

```
BEEP 2,0
```

то компьютер должен издать звук на высоте тона первой октавы на протяжение 2 секунд.

Звучание нот определяется при помощи двух цифр. Первая определяет длительность звука в секундах, а вторая — высоту звука в полутонах над «ДО» первой октавы. Таким образом, тональный код «ДО» первой октавы определяется цифрой «0». ДО — диез — 1, РЕ — 2 и т.д. до нот следующей октавы, которая определяется цифрой 12, потому что в октаве 12 полутонов. Вы можете продолжать до 13 и дальше: чем больше число, тем выше частота звука. Например:

```
BEEP 1,4 : BEEP 1,2 : BEEP 2,0
```

Вот простая программа для воспроизведения целой серии нот:

```
10 FOR X=0 TO 24
20 BEEP 2,X
30 NEXT X
```

Для нот, более низких, чем «ДО» первой октавы, число полутонов обозначается отрицательным числом.

Этим заканчивается вводная брошюра. Если вы чувствуете, что хорошо поняли ее, то предлагаем Вам прочесть «Руководство программиста», находящееся на магнитной ленте и содержащее развернутое описание языка БЕЙСИК компьютера «Синтез.2».

В приложении А приведено краткое описание операторов и функций языка БЕЙСИК.

ПРИЛОЖЕНИЕ А.

Краткое описание функций и операторов, реализуемых данной версией языка БЕЙСИК

ФУНКЦИИ

Функции — это зашитые в БЕЙСИК — систему подпрограммы, которые получая на входе одни значения, называемые аргументами, возвращают другие значения — результаты.

Функции используются в выражении простым включением в него имени функции с последующими аргументами.

Если в одном выражении используются и функции, и операции, то функции будут вычислены перед выполнением любых операций. Однако, этот порядок можно изменить, применяя скобки.

Принятые обозначения:

X, Y — аргументы функций;

A, B — операнды функций.

Все функции языка Бейсик сведены в следующую таблицу:

Имя функции	Тип аргумента	Действие (возвращаемое значение)
1	2	3
ABS	Число	Абсолютное значение
ACS	Число	Арккосинус в радианах, число должно быть в интервале от — 1 до 1
ASN	Число	Арксинус а радианах, число должно быть в интервале от — 1 до 1
ATN	Число	Арктангенс в радианах
ATTR	Два числовых аргумента X и Y, заключенных в скобки	Число, двоичный код которого представляет собой атрибуты Y-ой позиции X-ой строки экрана. Бит 7 (старший) равен 1 для мерцающего поля, и 0 для немерцающего. Биты с 5 по 3 — цвет фона. Биты с 2 по 0 — цвет закрашивания. 0<=X<=23 и 0<=Y<=31.
BIN		Это не обычная функция. За BIN записывается последовательность нулей и единиц, представляющая собой двоичное представление числа, которое записывается в память.
CHR$	Число	Символ, чей код представим числом X, округленным к ближайшему целому.
CODE	Строка символов	Код первого символа в строке X (или 0,если строка пустая)
COS	Число в радианах	Косинус X
EXP	Число	E в степени X
FN		FN с последующим именем, определенной пользователем функции (см. DEF). Аргументы должны заключаться в скобки. Даже если нет аргументов, скобки все равно должны записываться
IN	Число	Осуществляется ввод на уровне микропроцессора из порта X (0<=X<=FFFFH). Загружается пара регистров BC и выполняется команда ассемблера IN A, (C).
INKEY$	Нет	Чтение с клавиатуры. Возвращает символ, введенный с клавиатуры (в режиме [L] или [C]), если было действительное нажатие клавиши, или пустую строку в противном случае.
INT	Число	Округление к ближайшему меньшему целому
LEN	Строка символов	Длина строки
LN	Число	Натуральный логарифм, X>0.
NOT	Число	0, если X<>0; 1, если X=0. Операция имеет четвертый приоритет.

1	2	3
OR	Оба операнда числа	Логическая операция. 1, если B<>0 A OR B = A, если B=0 Операция имеет второй приоритет.
PEEK	Число	Значение байта в памяти по адресу X, округленному к ближайшему целому
PI	Нет	Число ПИ (3.14159265...)
POINT	Два числовых аргумента X и Y, заключенных в скобки	1, если точка экрана с координатами (X,Y) закрашена. 0, если эта точка имеет цвет фона. 0<=X<=255 и 0<=Y<=175.
RND	Нет	Очередное псевдослучайное число из последовательности, получаемой возведением в 75 степень модуля числа 65537, вычитаиием 1 и делением иа 65536. Число лежит в интервале 0<=Y<=1.
SCREEN$	Два числоаых аргумеита X и Y, заключенных в скобки;	Снмвол (обычный или иинверсиый), который появляется на экраие в строке X, позиии Y. Дает пустую строку, если символ ие опозиан.
SGN	Число	-1, если X<0 0, если X=0 1, если X> 0
SIN	Число в радианах	Синус X
SQR	Число	Кореиь квадратный. Ошибка, если X<0.
STR$	Число	Строка символов, которая должиа быть отображена, если X выводится
USR	Число	Вызывает подпрограмму в машиниых кодах, начальный адрес которой X. При возврате результатом будет содержимое регистровой пвры BC.
USR	Строка символов	Адрес группы байтов, задающих определёиный пользователем символ для закреплеиия его за X.
VAL	Строка символов	Вычисляет X как числовое выражение. Выдает сообщеиие об ошибке, если X содержит синтаксическую ошибку или дает строковое (не числовое) зиачеиие.
VAL$	Строка символов	Вычисляет X как строковое выражение. Выдает сообщение об ошибке, если X содержит синтаксическую ошибку или дает не строковое (числовое) значение.

ОПЕРАТОРЫ

Принятые обозначения:

A — одна буква;
V — переменная;
X, Y, Z — числовые выражения;
M, N — числовые выражения, которые округляются к ближайшему целому;
E — некоторое выражение;
F — выражение, имеющее строковое значение;
S — последовательность операторов, разделенных двоеточием «:»;
C — последовательность символов управления цветом. Каждый заканчивается запятой или точкой с запятой. Цветовой символ имеет форму операндов: PAPER, INK, FLASH, BRIGHT, INVERSE или OVER.

Текст произвольного выражения может располагаться в любом месте строки (за исключением номера строки, который должен размещаться в начале строки).

Все операторы, кроме INPUT, DEF и DATA могут использоваться и как команды и в программах.

Команда или строка программы может содержать несколько операторов, разделенных двоеточием «:».

Нет ограничений на положение оператора в строке, хотя есть некоторые ограничения в IF и REM.

Все операторы языка БЕЙСИК сведены в следующую таблицу:

Оператор	Действие оператора
1	2
BEEP X, Y	Воспроизводит звук длительностью X сек. и высотой Y полутоиов вверх от основного тона ДО (или вниз, если Y отрицательное)
BORDER M	Устанавливает цвет рамки (бордюра) экрана. 0<=M<=7
BRIGHT M	Устанавливает яркость выводимого символа: 0 — для обычиой яркости; 1 — для повышенной яркости; 8 — сохраняет существующую яркость.
CAT	Без MICRODRIVE не работает
CIRCLE X,Y,Z	Изображает дугу или окружность с центром в точке с координатами (X,Y) и радиусом Z.
CLEAR	Уничтожает все переменные и очищает занимаемую ими память Выполняет RESTORE и CLS, устанавливает PLOT позицию в нижнюю левую точку экрана и очищает GO SUB стек.
CLEAR N	Подобно CLEAR, но дополнительно изменяет системную переменную RAMTOP на 'N' и задает новый GO SUB стек.
CLOSE#	Без MICRODRIVE ие работает
CLS	(CLEAR SCREEN) очищает файл экрана
CONTINUE	Продолжает выполнение программы, начатой ранее и остановленной с сообщением, отличным от 0. Если было сообщение 9 или L то выполнение продолжается со следующего оператора. В других случаях с того оператора, где случилась ошибка.
COPY	Пересылает копию 22 строк экрана на приитер, если он подключен
DATA E1, E2, E3,...	Часть списка данных. Располагается только в программе.
DEF FNA(A1,...,AK)=E	Определяемая пользователем функция. Должна располагаться в программе. A,A1 и т.д. единственные буквы или буквы и $ для строковых аргументов, значений. Если нет аргументов, используется форма DEF FNA( ).
DELETE F	Без MICRODRIVE не работает
DIM A (N1,N2,...,NK)	Уничтожает массив с именем 'A' и устанавливает числовой массив 'A' с 'K' измерениями и присваивает всем его элементам значение 0.
DIM A$ (N1,N2,...,NK)	Уничтожает массив или строку с имеием 'A$' и устаиавливает символьный массив с 'K' измереннями и присваивает всем его элементам значение « ». Массив не определен до его описания в операторе DIM.
DRAW X, Y	То же самое, что и DRAW X, Y, 0. Чертит прямую линию
DRAW X,Y,Z	Изображает линию от текущей графической позиции в точку приращениями X,Y по дуге в Z радиан.
ERASE	Без MICRODRIVE не работает

1	2
FLASH N	Определяет, будет ли символ мерцающим или с постоянным свечением. N=0 для постоянного свечения, N=1 для мерцания. N=8 — для сохранения предыдущего состояния.
FOR A=X TO Y	FOR A=X TO Y STEP 1
FOR A=X TO Y STEP Z	Уничтожает скалярную переменную 'A' и устанавливает управляющую переменную 'X', предел 'Y', шаг приращения 'Z', зацикливает адрес, указанный и утверждении после FOR оператора. Проверяет, если начальное значение больше (если STEP>=0) или меньше (если STEP<0), чем предел, то происходит переход к утверждению NEXT A.
GO SUB N	Проталкивает строку с оператором GO SUB в стек для использования затем как GO TO N.
GO TO N	Продолжает выполнение программы со строки. 'N', если 'N' опущено, то с первой строки после этой.
IF X THEN S	Если 'X' истинно (не равно 0), то выполняется 'S', 'S' включает все операторы до конца строки.
INK N	Устанавливает цвет закрашивания (т.е. цвет, которым будут изображаться символы на цвете фона). 'N' в интервале от 0 до 7 указывает цвет. N=8 — оставить цвет без изменения, N=9 — увеличение контраста.
INPUT...	'...' есть последовательность вводимых символов, разделяемых как в операторе PRINT запятыми, точками с запятой или апострофами. Вводимыми символами могут быть: а) некоторый PRINT — символ, начинающийся не с буквы; б) имя переменной; в) строка имен переменных строкового типа. PRINT — символы в случае а) представляются так же, как и в операторе PRINT, за исключением того, что они все выводятся в нижнюю часть экрана. В случае б) компьютер останавливается и ждет ввода некоторого выражения с клавиатуры, значение которого будет присвоено переменной. Ввод осущесталяется обычным образом, а синтаксические ошибки выдаются мерцающим знаком [?]. Для строкового выражения вводной буфер устанавливается для размещения двух таких строк. Если первый вводимый символ STOP, то программа останавливается. Случай в) подобен случаю б) с той лишь разницей, что вводимая информация представляет собой строковый литерал неограниченной длины, и STOP в этом случае не сработает. Для останова надо нажать клавишу 'курсор вниз'.
INVERSE N	Символ управления инверсией выводимого символа. Если N=0, символ выводится в обычном виде с прорисовкой цветом закрашивания (INK) на цвете фона (PAPER). Если N=1, то цветовое решение изображения символа меняется на обратное.
LET V=E	Присваивает значение 'E' переменной 'V'. Ключевое слово LET не может быть опущено. Скалярная переменная не определена, пока не встретится в операторах LET, READ или INPUT. Если 'V' индексируемая строковая переменная или сечение строкового массива (подстрока), то присваивание осуществляется с усечением справа или дополнением пробелами до фиксированной длины.

1	2
LIST	То же, что и LIST0.
LIST N	Записывает текст программы в верхнюю часть экрана, начиная с первой строки, меньшей, чем 'N', и делает 'N' текущей строкой.
LLIST	То же, что и LLIST0.
LLIST N	Подобно LIST, но вывод осуществляется на принтер.
LOAD F	Загружает программу и переменные.
LOAD F DATA ( )	Загружает числовой массив
LOAD F DATA $ ( )	Загружает строковый массив.
LOAD F CODE M,N	Загружает старшие 'N' байтов, начиная с адреса 'M'.
LOAD F CODE M	Загружает байты, начиная с адреса 'M'.
LOAD F CODE	Загружает байты по адресу, с которого они были записаны на ленту.
LOAD F SCREEN$	Аналогично LOAD F CODE 16384, 6912. Очищает файл экрана и загружает его с кассетного магнитофона.
LPRINT	Подобно PRINT, но использует принтер.
MERGE F	Подобно LOAD F, но не затирает всю старую программу в памяти, а заменяет только те строки и переменные, у которых соападают номера или имена с такими же на ленте
MERGE F1,F2	Без MICRODRIVE не работает
NEW	Запускает по новой систему программирования БЕЙСИК, уничтожая старую программу и переменные и используемую память.
NEXT A	а) находит управляющую переменную 'A'; б) прибавляет к ней значение STEP; в) если STEP>=0, а значение 'A' стало больше значения 'предел', или STEP<0, а значение 'A' меньше, чем значение 'предел', то происходит переход к оператору цикла.
OPEN#	Без MICRODRIVE не работает
OUT M,N	Выводит байт 'N' в порт 'M'. Операции выполняется на уровне микропроцессора (загружает в регистровую пару BC адрес 'M', а в регистр A — 'N' и выполняет команду ассемблера OUT (C), A . 0<=M <=65535, —255<=N<=255.
OVER N	Управляющий символ надпечатывания по выведенной строке. Если N=0, то выводимый символ затирает существующий в данной позиции. Если N=1, то новый символ соединяется со старым, образуя закрашивающий цвет, при условии, что старый символ имел указание цветв, отличное от старого, или цвет фона, если оба указывают на один и тот же цвет (либо фона, либо закрашивания).
PAPER N	Подобен INK, но управляет цветом фона.

1	2
PAUSE N	Останавливает выполнение программы и задерживает изображение на экране на 'N' кадров (50 кадров в сек. — частота кадровой развертки) или до нажатия любой клавиши. 0<=N<=65535. При N=0 время задержки не учитывается и продолжается до первого нажатия клавиши.
PLOT C; M,N	Выводит точку закрашивающего цвета (обработанная OVER и INVERSE) с координатами (ABS (M), ABS(N)), смещает графическую позицию. Если цветовой символ 'C' не специфицирован иначе, то цвет закрашивания в позиции, где расположена эта точка, изменяется на текущий сплошной закрашивающий цвет, и другие указания (цвет фона, мерцание, яркость) остаются без изменения. 0<=ABS(M) <=255 0<=ABS(N) <=175
POKE M,N	Записывает значение 'N' в байт памяти по адресу 'M'. 0<=M <=65535, — 255 <=N <=255
PRINT..	'...' последовательность PRINT — символов, разделенных запятыми, точками с запятой или апострофами, которые выводятся в экранный файл для отображения на экране телевизора. Точка с запятой сама действия не вызывает, а используется для разграничения символов. Запятая порождает управляющий символ «запятая», а апостроф порождает символ ENTER В конце оператора PRINT, если он не заканчивается точкой с запятой, запятой или апострофом, автоматически выводится символ ENTER PRINT — символом может быть: а) пустая строка; б) числовое выражение. Если значение выражения отрицательно, то выводится знак минус. Если X<=10**(5) или X>=10**13, вывод осуществляется в показательной форме. Мантисса представляется 8-мью цифрами (с нормализацией) и десятичной точкой (отсутствует только тогда, когда в мантиссе одна цифра) после первой цифры. Показатель степени записывается после буквы 'E' с последующим знаком и двумя цифрами порядка. Иначе X выводится как обычное десятичное число с 8-мью значащими цифрами; в) строковое выражение. В строке возможны пробелы до и после символов. Управляющие символы вызывают определяемое ими действие. Не отображаемые на экране символы выводятся как '7'; г) AT M,N Вывод в позицию 'N' строки 'M'; д) TAB N Перемещает позицию вывода в столбец N текущей строки; е) цветовой символ в форме PAPER, INK, FLASH, BRIGHT, INVERSE или OVER оператора.
RANDOMIZE	То же, что и RANDOMIZE 0

1	2
RANDOMIZE N	Устанавливает системную переменную SEED, используемую для вычисления очередного значения функции RND. Если N< >0, то SEED принимает значение 'N'. Если N=0, то SEED принимает значение другой системной переменной FRAMES, подсчитывающей кадры, отображаемые на экране, что обеспечивает вполне случайное число. N должно лежать в интервале от 0 до 65535.
READ V1,V2,..., VK	Присваивает переменным V1,...VK одной за другой значения, последовательно представленные в списке DATA.
REM...	Вставка комментариев. Не выполняется. '...' может быть последовательностью символов (исключая ВК).
RESTORE	То же самое, что и RESTORE O.
RESTORE N	Перезаписывает указатель данных в первый оператор DATA в строке меньшей, чем N. Следующий оператор READ начнет считывание отсюда.
RETURN	Возврат из подпрограммы, ссылается на оператор GO SUB в стеке и передает управление на строку после него.
RUN	То же самое, что и RUN O.
RUN N	Запуск выполнения БЕЙСИК-программы. Выполняет CLEAR, а затем GO TO N.
SAVE F	Вывод на ленту программ и данных.
SAVE F LINE M	Записывает на ленту программу и переменные таким образом, что при загрузке программа автоматически выполняется со строки 'M'.
SAVE F DATA ( )	Запись на ленту числового массива.
SAVE F DATA $( )	Запись на ленту строкового массива
SAVE F CODE M,N	Запись на ленту 'N' байтов, начиная с адреса 'M'.
SAVE F SCREEN$	Аналогично SAVE F CODE 16384, 6912
STOP	Останавливает выполнение программы. Продолжение (CONTINUE) будет осуществляться со следующего оператора.
VERIFY	То же, что и LOAD, за исключением того, что данные не загружаются в ОЗУ, а сравниваются с находящимися там кодами.